

(19) World Intellectual Property Organization  
International Bureau



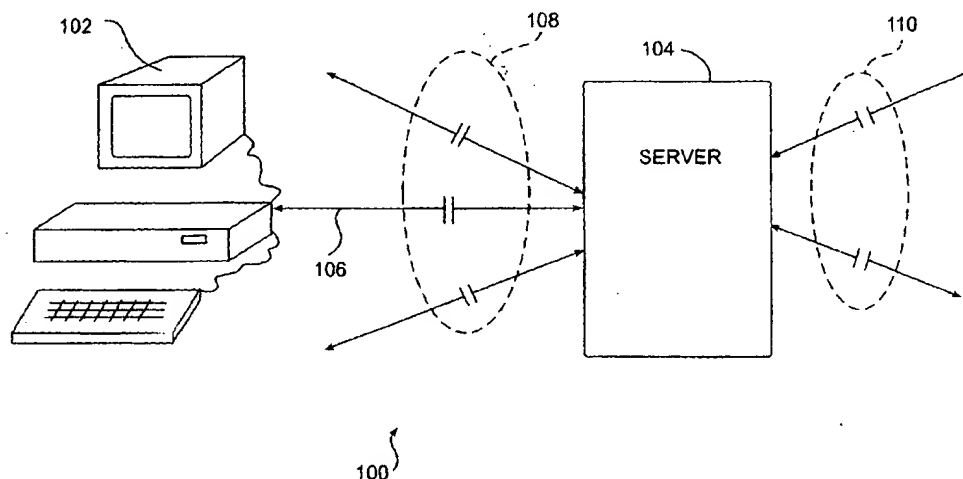
(43) International Publication Date  
25 October 2001 (25.10.2001)

PCT

(10) International Publication Number  
WO 01/80086 A2

- (51) International Patent Classification<sup>7</sup>: G06F 17/30
- (21) International Application Number: PCT/US01/12146
- (22) International Filing Date: 13 April 2001 (13.04.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/548,760 13 April 2000 (13.04.2000) US
- (71) Applicant: NEOPLANET, INC. [US/US]; 1524 W. Street, Suite 102, Tempe, AZ 85281 (US).
- (72) Inventors: FRISKEL, James, P.; 3221 S. Atlantic Avenue #704, Cocoa Beach, FL 32931 (US). NOE, Michael, J.; 9756 Turquoise Avenue, Scottsdale, AZ 85258 (US).
- (74) Agent: BOSWELL, Mary, Jane; Morgan, Levis & Bookies LLP, 1800 M Street, N.W., Washington, DC 20036-5869 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:  
— without international search report and to be republished upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR A DUAL RENDERING ENGINE BROWSER



(57) Abstract: An apparatus for processing HTML text is disclosed. The apparatus includes a computer having a processor and memory that is operable to run a parser program. The parser program includes two or more rendering engines. The parser program is operable to receive HTML formatted text, select a rendering engine; and process the HTML text.

**METHOD AND APPARATUS FOR A DUAL RENDERING ENGINE BROWSER**

**Inventors:** James R. FRISKEL and Michael J. NOE

**5    RELATED APPLICATIONS**

This application is related to U.S. Utility Application Serial Number 09/548,760, filed April 13, 2000, which is herein incorporated by reference in its entirety.

**TECHNICAL FIELD OF THE INVENTION**

10        This invention relates to the field of computer programs and more specifically to a method and apparatus for a dual rendering engine browser.

**BACKGROUND**

15        The usage of the INTERNET is increasing at a very large rate. Generally, a user accesses content on the Internet by executing a "web browser" or "parser" program locally at a client (such as a home computer) and interconnecting to a server of an Internet Service provider. The interconnection may use any of a variety of communication links such as a dial-up telephone communication link or a dedicated  
20        communication link, such as a cable modem. The web browser is a computer program that allows the client to exchange information with the remote server. Any of a variety of web browsers are available, such as NETSCAPE NAVIGATOR, MICROSOFT EXPLORER and others that allow users to conveniently access and navigate the Internet using a graphical user interface. Web browsers receive web pages in a format or  
25        language understandable by a web browser, such as Hypertext Markup Language ("HTML"). HTML is a common markup language that describes the structure and behavior of a document. Web browsers, through the use of a rendering engine, interpret the web pages and generate a corresponding display of the web pages using a graphical user interface. One drawback of the present invention is that different rendering engines  
30        interpret HTML differently. Therefore, the output of various web browsers may differ. Currently, to view different outputs, one must use different web browsers at separate

times. This is cumbersome to both users of web browsers and developers of web content who need to know what their work will look like using different browsers.

### **SUMMARY OF THE INVENTION**

5

From the foregoing it may be appreciated that a need has arisen for a method and apparatus for a dual rendering engine browser. In accordance with the present invention, a method and apparatus for a dual rendering engine browser is provided that allows a user to view a document formatted in a markup language with a rendering engine of the user's choice or using more than one rendering engine.

10

According to an embodiment of the present invention, there is provided an apparatus for processing HTML text. The apparatus includes a computer having a processor and memory that is operable to run a parser program. The parser program includes two or more rendering engines. The parser program is operable to receive HTML formatted text, select a rendering engine; and process the HTML text.

15

A technical advantage of the present invention includes the ability of a user to choose what rendering engine best fits the particular document to be viewed. Another technical advantage of the present invention includes the ability to use multiple rendering engines simultaneously. Other technical advantages are readily apparent to one skilled in the art from the following figures, description, and claims.

20

### **BRIEF DESCRIPTION OF THE DRAWINGS**

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts, in which:

25

FIG. 1 is a diagram illustrating a network interconnection including a client and an Internet server;

30

FIG. 2 is a block diagram illustrating the client and a remote Internet server;

FIG. 3 is a block diagram showing the functional parts of the program of the present invention;

FIG. 4a and FIG. 4b illustrate web browser display in accordance with the teachings of the present invention; and

5        FIG 5 is a flowchart illustrating the operation of an embodiment of the present invention.

### **DETAILED DESCRIPTION OF THE INVENTION**

10        Turning first to the nomenclature of the specification, the detailed description which follows is represented largely in terms of processes and symbolic representations of operations by conventional computer components, including a central processing unit ("CPU") or processor associated with a general purpose computer system, memory  
15        storage devices for the CPU, and connected pixel-oriented display devices. These operations include the manipulation of data bits by the CPU and the maintenance of these bits within data structures resident in one or more of the memory storage devices. Such data structures impose a physical organization upon the collection of data bits stored within computer memory and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of  
20        computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

      In addition, it should be understood that the programs, processes, methods, etc. described herein are but an example of one implementation of the present invention and are not related or limited to any particular computer, apparatus or computer language.  
25        Rather, various types of general purpose computing machines or devices may be used with programs constructed in accordance with the teachings described herein. Similarly, it may prove advantageous to construct a specialized apparatus to perform the method steps described herein by way of dedicated computer systems with hardwired logic or programs stored in non-volatile memory, such as read only memory.

30        Referring now in more detail to the drawings, FIG. 1 is a diagram illustrating a network interconnection 100 including a client 102 and an Internet server 104. Network

interconnection 100 includes the interface between Internet server 104 and a plurality of clients through a plurality of direct communication lines 108. For example, network interconnection 100 includes an interconnection between client 102 and Internet server 104 through a direct communication line 106. Network interconnection 100 also  
5 includes the interface between Internet server 104 and other Internet servers through a plurality of communication lines 110. In this manner, Internet server 104 allows the plurality of clients directly interconnected with server 104, such as client 102, to communicate with other Internet users by providing an interconnection to other Internet servers through communication lines 110. Other Internet users are also directly  
10 interconnected with a server such that a communication path may be established between one client and another client through the various serves of the Internet. Internet server 104 is a computer such as a personal computer, file server, workstation, minicomputer, mainframe, or any other computer capable of communicating and interconnecting with other computers. Internet server 104 will preferably include a processor, a printer, an  
15 input device such as a mouse and/or a keyboard, a monitor, a floppy disk drive, memory, a modem, and a mass storage device such as a hard disk drive. Communication lines 110 and direct communication lines 108 may be any type of communication link capable of supporting data transfer. For example, these communication lines may include any combination of an Integrated Services Digital Network ("ISDN") communication line, a  
20 hard wired line, or a telephone link.

Client 102 may be similar to Internet server 104 and may be implemented using virtually any type of computer. Client 12 will preferably be a personal computer having a processor, a printer, an input device such as a mouse and/or a keyboard, a monitor, a floppy disk drive, memory, a modem, and a mass storage device such as a hard disk  
25 drive. Client 102 and Internet server 104 will be operating under the control of an operating systems such as MS-DOS, Macintosh OS, WINDOWS NT, WINDOWS 95/98/2000, OS/2, UNIX, LINUX and the like. Client 102 and Internet server 104 may execute any number of available application programs such as a web browser.

In operation, the various clients of network interconnection 100, such as client  
30 102, may communicate through server 104 with any other client connected to the Internet. For example, client 102, generally using a web browser application program,

may couple to Internet server 104 and provide the address of URL of an Internet web page. The Internet web page will generally be stored on another Internet server located remotely. Internet server 104, through the plurality of communication lines 110, communicates with other Internet servers. Eventually, the receiving Internet server is  
5 found and in response, transmits the HTML formatted Internet web page back to Internet server 104 for viewing by client 102 using a web browser.

Although network interconnection 100 has been illustrated and described in FIG. 1 as being a node or interconnection on the Internet, network interconnection 100 may be any interconnection found on any computer network such as a local area network  
10 ("LAN"), a wide area network ("WAN"), an intranet, such as a corporate intranet, or any other communications and data exchange system created by connecting two or more computers. The present invention will be illustrated and described with an implementation using the Internet, however, it should be understood that the present invention is not limited to only implementations using the Internet. It may, for example,  
15 be implemented on a single computer operating by itself.

FIG. 2 is a block diagram illustrating client 102 and a remote Internet sever 104. Client 102 includes a processor 208, input/output ("I/O") devices 210, and a client memory 202. Processor 208, under the control of an operating system 204, controls the operation of client 102 and is used to retrieve, process, store, and display data. Operating  
20 system 204 and a parser program 206 are stored in client memory 202. Client memory 202 may be a random access memory ("RAM"), Read-only memory ("ROM"), or other types of memory.

Processor 208 is typically implemented as a microprocessor, such as those manufactured by INTEL, MOTOROLA, or ADVANCED MICRO DEVICES.  
25 Processor 208 may include an arithmetic logic unit to assist in performing mathematical operations. Processor 208 communicates control, address, and data signals with operating system 204 and with the remaining components of client 102 through a system bus. Processor 208 interprets and executes instructions that have been fetched or retrieved from client memory 202 and may be implemented as a single integrated circuit  
30 or as a combination of integrated circuits.

I/O devices 212 may include any peripheral that allows data to be exchanged with client 102 and may include such devices as a keyboard, a monitor, a printer, a modem, a pointing device, such as a mouse, a floppy disk drive, a mass storage device, such as a hard disk drive, and the like. The mass storage device is used to store computer files including application programs and data files. For example, the mass storage device may be used to store parser program 206 and operating system 204.

Operating system 204 includes a set of computer programs that control the internal functions of client 102, thereby allowing client 102 to run application programs. Operating system 204 is typically stored in and provided from a mass storage device, such as a hard disk drive, a floppy disk drive, a CD ROM drive or a ROM chip. During start-up or initialization of client 102, operating system 204 is loaded into client memory 202. Application programs, such as parser program 206, may also be loaded into client memory 202 along with operating system 204.

Parser program 206, in one embodiment, is a graphical web browser that allows a user to view images, fonts, and document layouts provided in a web page by converting large units of data into smaller, more easily interpreted, units of data. Parser program 206 reads the tagged document of a web page or other markup language formatted document, such as those provided in Hypertext Markup Language (HTML) format. HTML uses tags to identify the parts of a web page, such as headings, bulleted lists, body text, on-screen forms, including fill-in text boxes, options buttons, radio buttons, and drop-down list boxes, images to be displayed, hypertext links, colors, font, and various other formatting tags. Parser program 206 formats the various parts of the document for on-screen display as directed by the HTML tags. It accomplishes this by using a rendering engine which interprets the HTML formatted document. In the present invention, a web browser includes multiple rendering engines. Rendering engines are discussed in greater detail in conjunction with FIGURES 3 through 5. While HTML is used as an example markup language, the present invention is not limited to HTML formatted documents but includes other markup languages as well, such as SGML and GML.

In the present invention, a user is able to choose a rendering engine to process the markup language formatted documents information from a selection of two or more

rendering engines such as the Internet Explorer rendering engine used in the Internet Explorer web browser or the MOZILLA rendering engine used in the Netscape browser as well as any other rendering engines. Typically, rendering engines for different web browsers can be called by third party software by specifying an application ID and an interface ID. The use of, and accessing of, rendering engines is known to those skilled in the art. Alternatively, a user can view a markup language formatted document processed by different rendering engines simultaneously in different windows of the display.

In another embodiment, parser program 206 can also be used as part of a program or series of programs used to develop markup language formatted documents such as web pages. By including the different rendering engines from different web browsers, a developer using the parser program 206 of the present invention will be able to ensure the markup language formatted document under development will work with different browsers. Further discussions of these embodiments are presented in the following descriptions.

Remote Internet server 104 includes a server memory 222, a processor 224, and a mass storage device 226. Remote Internet server 105 also preferably includes various I/O devices 228 such as those mentioned above in the description of I/O devices 212 of client 102.

Operating system 220 and a web server 222 are stored in mass storage device 226 and are shown loaded into server memory 223. Processor 224, in conjunction with operating system 220, controls the operation of remote Internet server 104. Processor 224 fetches and executes various instructions stored in server memory 223. Operating system 220 operates similarly to operating system 204 of client 102 and includes a set of computer programs that control the internal functions of remote Internet server 104. Operating system 220 controls the allocation and usage of hardware resources such as server memory 223, processor 224, mass storage device 226, and optional I/O devices 228. Operating system 220 may be any of a variety of available operating systems depending partially on the hardware of server 104. For example, operating system 220 may be implemented as MS-DOS, the Macintosh OS, OS/2, WINDOWS NT, WINDOWS NT/ WINDOWS 95/98/2000, UNIX, LINUX or any of a variety of other operating systems.



Web server program 220 is an application program shown loaded into server memory 223. Web server program 220 is stored in mass storage device 226 and is then loaded into server memory 223 when selected by a user or is initiated on the server side. In response to a client request, markup language formatted documents will be provided to parser program 206.

FIGURE 3 illustrates a block diagram showing the functional parts of the program of the present invention. Illustrated is a parser program 304 having a first rendering engine 306 and a second rendering engine 308 which is coupled to a display system 310. In operation, parser program 304 is operable to receive text files written in a markup language, such as HTML formatted documents. Depending on the uses of parser 304, HTML formatted documents 302 can be loaded via a request from parser 304 to send files from a remote server, or a parser 304 can retrieve HTML formatted documents 302 from a local storage device. After receiving HTML formatted documents 302, a choice is then made as to which rendering engine is to be used. This choice can be made by a user operating the program. For example, a user may choose that a first rendering engine 306 is to be used. In that case, the HTML document 302 received by parser 304 will be processed using first rendering engine 306. Then the rendered text can be displayed by display routine 310 on a monitor or other display device. On the other hand, if the second rendering engine 308 is to be used, then the HTML documents 302 is processed by second rendering engine 308 and then displayed by display routine 310. Of course, there are times when both rendering engines are to be used so as to view the web text side-by-side or rendered by two different rendering engines in order to determine any differences introduced by the different rendering engines.

In one embodiment, the rendering engines 306 and 308 can be the Internet Explorer rendering engine by Microsoft and/or the Mozilla rendering engine by Netscape. Rendering engines are operable to perform such tasks as format a document for display based on a markup language as well as manage user interactions and host associated elements such as ActiveX controls and JAVA applets. Parser program 304 can be any program operable to receive hypertext files and prepare them for sending on to a rendering engine. Display 310 is a program and/or hardware operable to display the formatted HTML files.

FIGURE 4a illustrates a sample display of a web browser implementing the present invention. Web browser 400 contains a multitude of windows including a navigational window 402, a channel window 404, an information window 406 and a main display window 408. In operation, navigation window 402 will contain  
5 information on navigating a web browser such as forward-backward buttons, search buttons, etc., as is commonly well-known in the art. Channel window 404 contains a list of sites of interest that are organized via topic. Information window 406 can be provided with live, real-time advertisements or banner advertising. Display window 408 is used to display the requested web page after processing by a web rendering engine. In this  
10 embodiment, a pull-down menu located on navigation window 402 or elsewhere could include a choice between rendering the HTML formatted hypertext page using a first rendering engine or a second rendering engine, or to use both.

FIGURE 4b illustrates a web browser in which web display window 408 has been divided into a first window for 408a and a second window for 408b. This embodiment is  
15 useful in cases where the choice is made to display one HTML formatted text, i.e., a web page, using both of the rendering engines simultaneously. In that case, the output from a first rendering engine would be displayed in a first window 408a and the output from a second rendering engine would be displayed in a first window 408a and the output from a second rendering engine would be displayed in a second window 408b. This same  
20 concept can be extended to cases where more than two rendering engines.

FIGURE 5 illustrates a method for displaying markup language formatted documents using one or more rendering engines. In step 502, a parser program is initiated. By parser program, this could be a web browser program operable to receive web pages written in HTML language from a remote computer via a connection such as  
25 a phone connection or a direct connection between the computers. The parser program can also be a program used to translate HTML documents located locally on a single computer. The parser program can also be a development tool used to develop HTML documents. Starting the parser program also initiates the transfer of HTML formatted documents to the parser program. Next, in step 504, an initial rendering engine is  
30 selected. This can be done automatically by the software which could have a default rendering engine or it could be done by a user who initially selects a default initial

rendering engine by setting a preference or in some similar fashion. Then, in step 506, a user can choose what rendering engine it wants to use. In step 506, a user can choose to either stay with the rendering engine currently being used, change rendering engines, or use multiple rendering engines. If a user chooses to use the same rendering engine, then, in step 508, the HTML document is displayed using that rendering engine. If the user decides to change rendering engines, then a new rendering engine id is sent by the parser program in step 510, initiating the use of the new rendering engine. Typically, each unique rendering engine will have a unique engine ID. Of course, in the present invention any means of accessing a rendering engine, either one publicly available or a proprietary engine, can be accomplished by the present invention. In step 512, the result is displayed. Finally, if a user chooses to use multiple rendering engines, then all rendering engines ids are sent to initiate the rendering engines in step 513. In step 514, the multiple rendering engine results are displayed. In this manner, comparisons can be made as to what rendering engine most efficiently renders the HTML document.

Thus, it is apparent that there has been provided, in accordance with the present invention, a method and apparatus for interpreting HTML formatted documents using multiple rendering engines that satisfies the advantages as set forth above. Although, the preferred embodiment has been described in detail, it should be understood that various changes, substitutions, and alterations can be made herein. For example, the present invention may be implemented on any computer, computer network, or interconnection of computers, as well as any standalone computer. Although the present invention has been described and shown as an element using the Internet, the scope of the present invention is not limited to the Internet. Other examples of changes, substitutions and alterations are readily assertable by one skilled in the art and could be made without departing from the scope and spirit of the present invention.

**WHAT IS CLAIMED IS:**

1. A method for processing markup language formatted documents comprising:
  - receiving the document;
  - providing two or more rendering engines;
  - 5 selecting a rendering engine; and
  - processing document with the selected rendering engine.
2. The method of Claim 1, wherein the step of selecting further comprises:
  - selecting two or more rendering engines, and
  - processing the document with the rendering engines.
3. The method of Claim 1, further comprising the step of displaying the processed document.
4. The method of Claim 2, further comprising the step of displaying the processed document in multiple windows.
5. An apparatus for processing markup language formatted documents comprising:
  - a computer having a processor and memory, the computer operable to run a parser program, the parser program comprising two or more rendering engines, the parser
  - 5 program operable to:
    - receive documents;
    - select a rendering engine; and
    - process the documents.
6. The apparatus of Claim 5, wherein the parser program is operable to display the documents.
7. The apparatus of Claim 5, wherein the parser program is further operable to allow a user to select two or more rendering engines.
8. The apparatus of Claim 5, wherein the parser program is operable to display a first window containing a document processed by a first rendering engine and to display a second window containing a document rendered by a second rendering engine.

9. The apparatus of Claim 1, wherein the markup language formatted documents are received from a remote server.

10. The apparatus of Claim 5, wherein the parser program is a web browser.

11. The apparatus of Claim 5, wherein the parser program is a web page development program.

12. A computer-readable medium containing instructions for markup language formatted documents, by:

reading in the documents;

processing the documents by selecting one or more rendering engines from a plurality of

5 rendering engines.

13. The medium of Claim 12, wherein the instructions further comprises allowing a user to select two or more rendering engines.

14. The medium of Claim 12, wherein the instructions further comprise displaying the documents.

15. The medium of Claim 13, wherein the instructions further comprise displaying a first window containing a document processed by a first rendering engine and displaying a second window containing a document rendered by a second rendering engine.

1/5

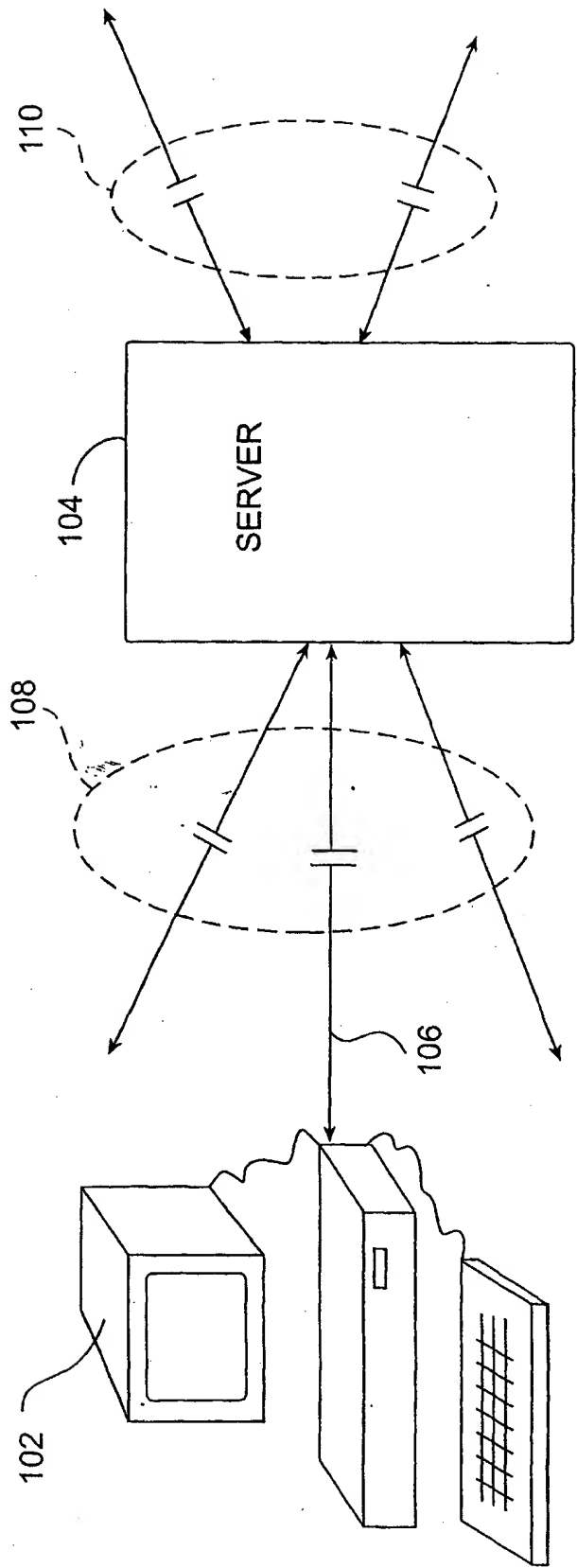
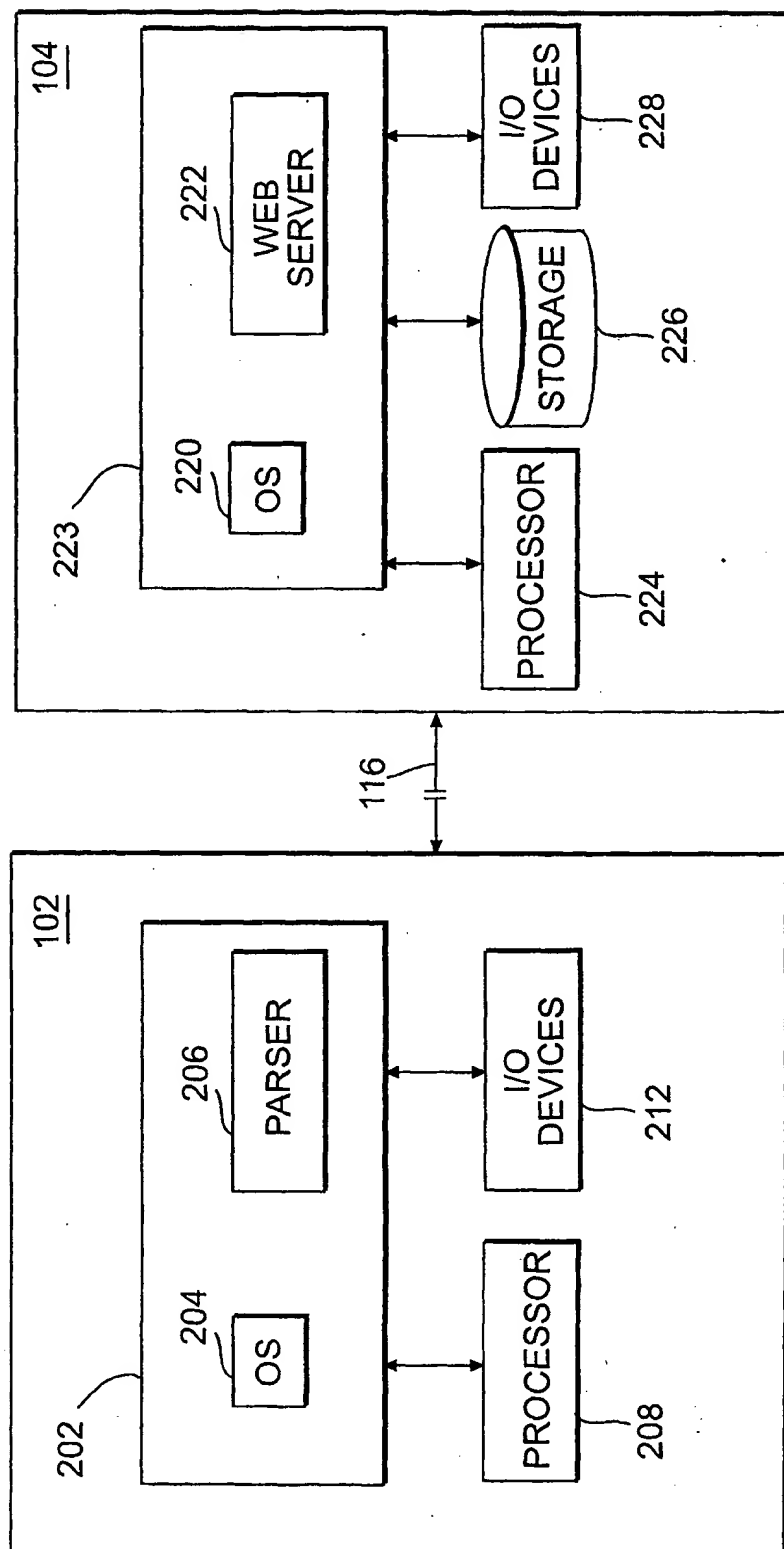


FIG. 1

2/5

**FIG. 2**

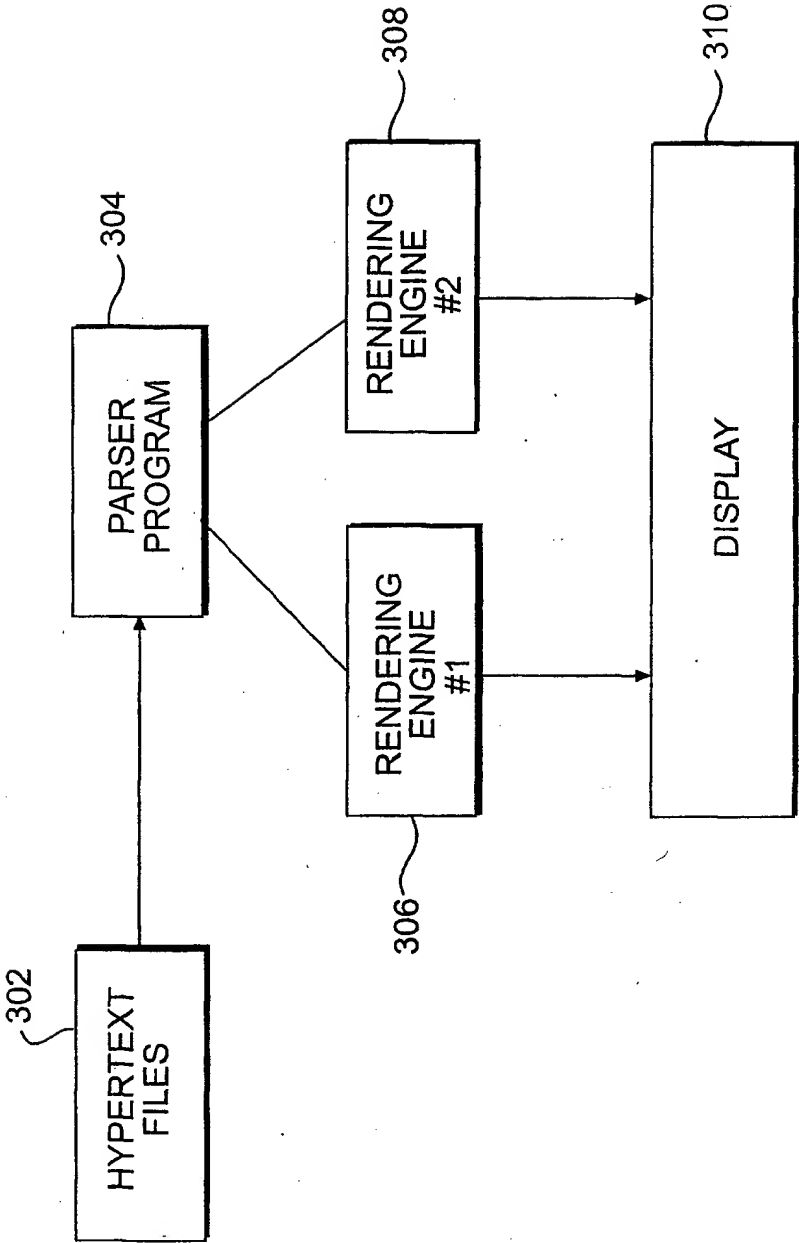
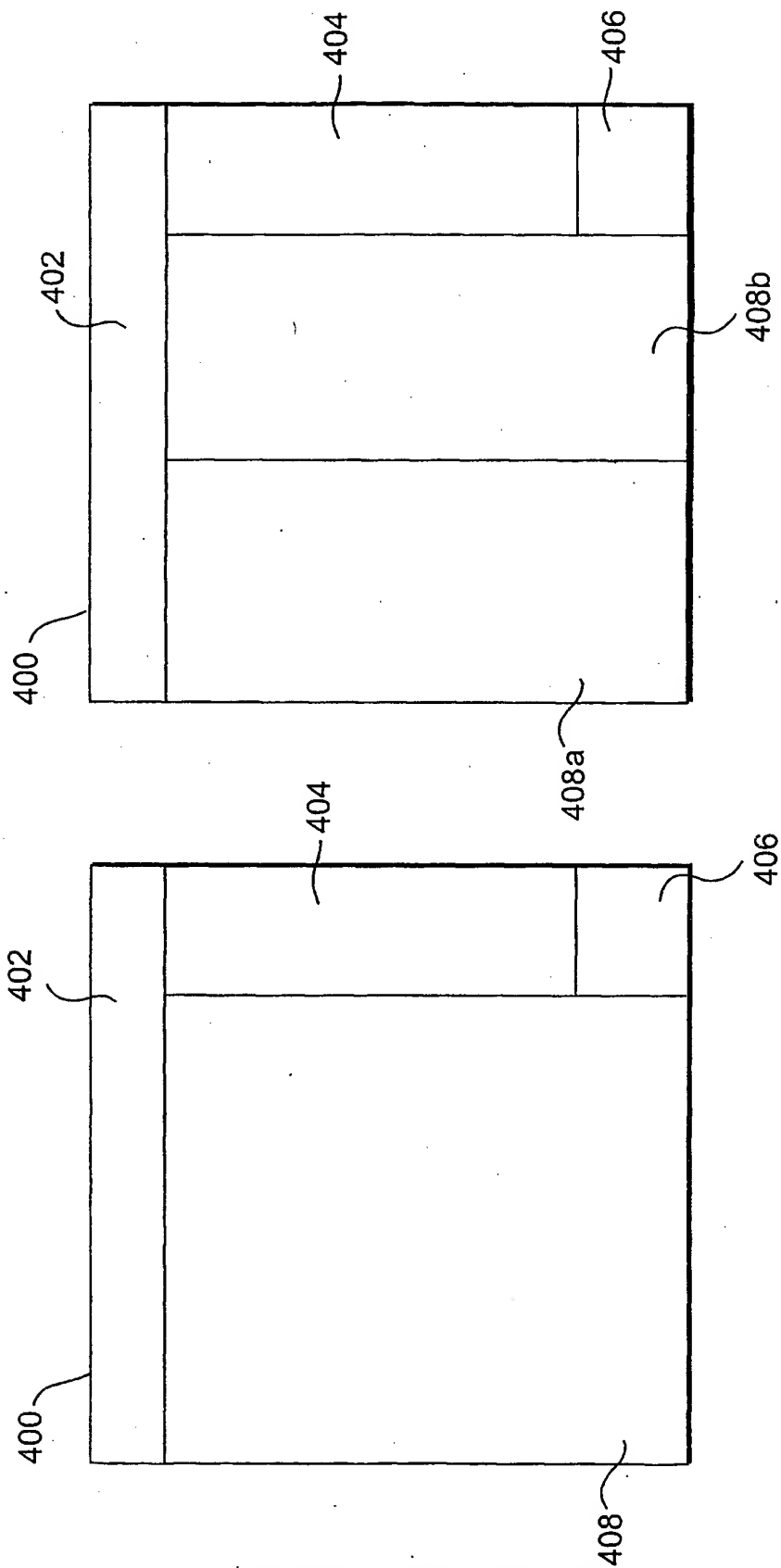


FIG. 3



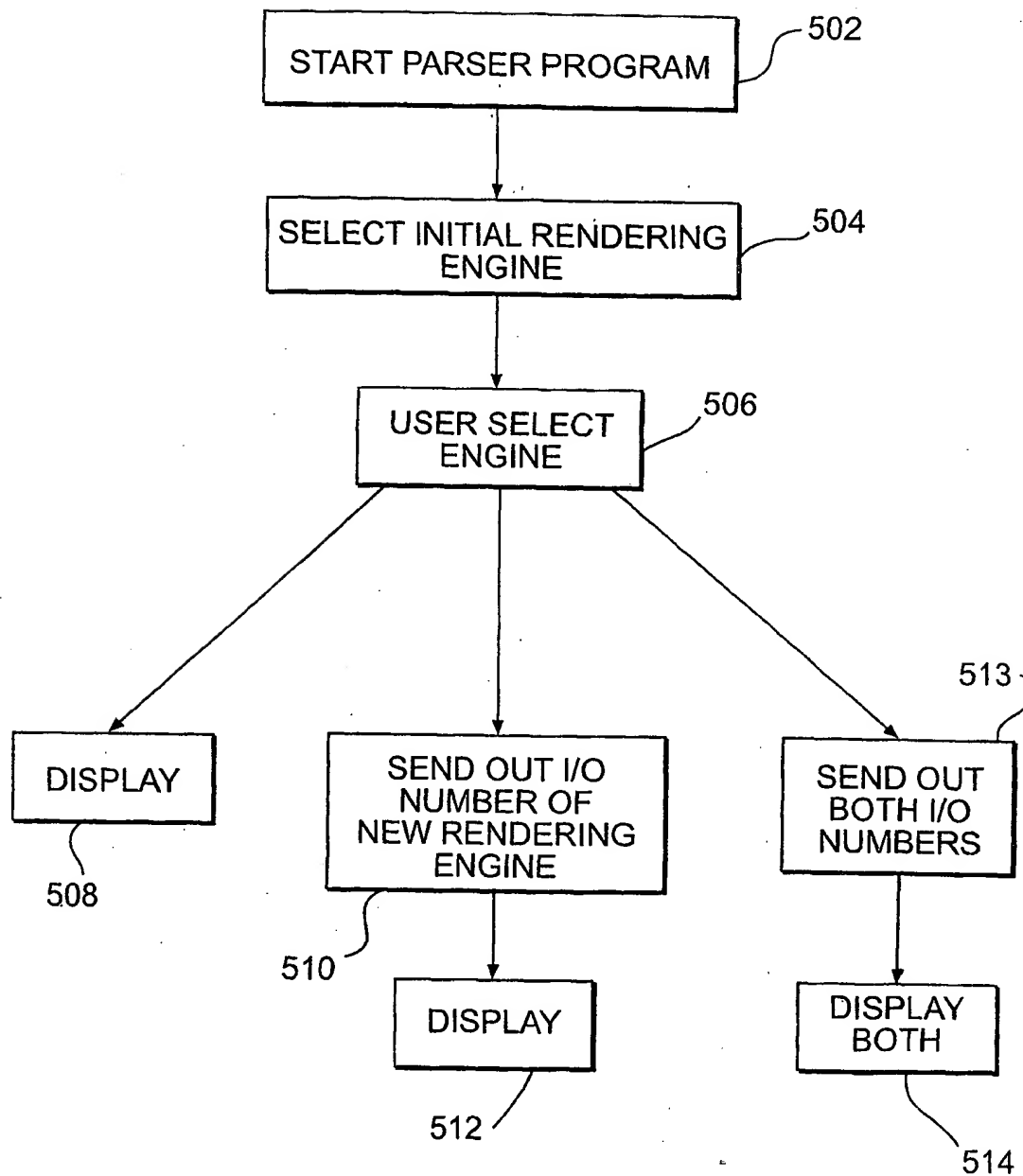
4/5



**FIG. 4B**

**FIG. 4A**

5/5

**FIG. 5**